Invited paper for a special issue of the
International Journal on Artificial Intelligence Tools, (IJAIT) 1999.

PREPROCESSING AND INTEGRATION OF DATA FROM MULTIPLE SOURCES FOR KNOWLEDGE DISCOVERY*

MARION G. CERUTI
Space and Naval Warfare Systems Center - San Diego, Code D4121
53560 Hull St., San Diego, CA, 92152-5001, USA
E-mail: ceruti@spawar.navy.mil

MAGDI N. KAMEL
Department of Systems Management, Naval Postgraduate School
555 Dyer Rd., Monterey, CA, 93943, USA
E-mail: kamel@nps.navy.mil

The explosive growth in the generation and collection of data has generated an urgent need for a new generation of techniques and tools that can assist in transforming these data intelligently and automatically into useful knowledge. Knowledge discovery is an emerging multidisciplinary field that attempts to fulfill this need. Knowledge discovery is a large process that includes data selection, cleaning, preprocessing, integration, transformation and reduction, data mining, model selection, evaluation and interpretation, and finally consolidation and use of the extracted knowledge. This paper addresses the issues of data cleaning and integration for knowledge discovery by proposing a systematic approach for resolving semantic conflicts that are encountered during the integration of data from multiple sources. Illustrated with examples derived from military databases, the paper presents a heuristics-based algorithm for identifying and resolving semantic conflicts at different levels of information granularity.

Keywords: Command and control, database integration, data mining, data warehousing, knowledge discovery, semantic heterogeneity

1. Introduction

The explosive growth of government, business, and scientific databases has overwhelmed the traditional, manual approaches to data analysis and created a need for a new generation of techniques and tools for intelligent and automated knowledge discovery in data. The field of knowledge discovery, an emerging and rapidly evolving field that draws from other established disciplines such as databases, applied statistics, visualization, arti-

* This work was produced by U. S. government employees in the course of their employment and is not subject to copyright. A preliminary version of this work was presented at the IEEE Knowledge and Data Engineering Exchange Workshop, KDEX'98.[1]

ficial intelligence and pattern recognition, specifically focus on fulfilling this need. The goal of knowledge discovery is to develop techniques for identifying novel and potentially useful patterns in large data sets. These identified patterns typically are used to accomplish the following goals:

- to make predictions about new data,
- to explain existing data
- to summarize existing data from large databases to facilitate decision making, and
- to visualize complex data sets.

Knowledge discovery is as an interactive and iterative process that consists of a number of activities for discovering useful knowledge.[2] The core activity in this process is data mining, which features the application of a wide variety of algorithms to discover useful patterns in the data. Whereas most research in knowledge discovery has concentrated on data mining, other activities are as important for the successful application of knowledge discovery. These include data selection, data preparation, data cleaning, and data integration. After data-mining algorithms are applied, additional activities are essential to ensure that useful knowledge is derived from the data. One such activity is the proper interpretation of the results of data mining.

This paper addresses the activities of data cleaning and integration, as important steps in the knowledge discovery process. Data of high quality are required for a successful data-warehouse environment because poor data quality may have catastrophic impacts on decision making.[3 and 4] Specifically, this paper covers the issues of semantic heterogeneity, such as conflict identification and resolution, for data integration from multiple sources. It cites examples of semantic heterogeneity derived from databases of United States Department of Defense (DOD) tactical systems; however, the concepts can apply to other types of information-systems applications.

Although a total solution to the problem of semantic integration is computationally intractable, a partial solution focused on specific classes of inconsistencies is offered here. Specifically, a three-phased methodology is presented for identifying and resolving semantic heterogeneity. The algorithm of this methodology is depicted in a series of trouble-shooting flow charts that consider semantic heterogeneity on various levels.

The paper is organized as follows. Section 2 reviews the concepts of semantic heterogeneity and presents a classification based on information granularity. Section 3 presents examples of case studies in semantic heterogeneity from database-integration efforts in the area of Command, Control, Communications, Computers and Intelligence (C^4I). Section 4 develops the algorithm for identifying and resolving semantic heterogeneity at three levels in the conflict-resolution process. Section 5 presents some conclusions and provides a discussion of directions for future research.

2. Semantic Heterogeneity

Sheth and Larson define semantic heterogeneity as the existence of disagreement about the meaning, interpretation, or intended use of the same or related data.[5] Semantic heterogeneity can be classified broadly into two categories, schema and data. Schema con-

flicts include homonyms and synonyms, as well as differences in data types, length, units of measure, and levels of object abstraction. Schema conflicts such, as homonyms and synonyms, can be determined at schema-definition time. Other schema conflicts, including differences in data domains and units of measure can be determined from the schema definition time when this information is specified as part of the attribute name. (For examples, see subsection 4.3.)

Some semantic inconsistencies can be discovered in multiple ways. However, data conflicts are best discovered and can be verified only at run time using queries against various database components. Data-fill heterogeneity includes different units of measure, different levels of precision, different data values for the same measurement, etc. We continue the heterogeneity classification process to include three distinct, but related levels of semantic heterogeneity.

Fig. 1 shows the classification of semantic conflicts. This classification was chosen because it contributes to a logical progression to simplify and facilitate the development of the algorithm described in section 4, which is a blueprint for the systematic identification and resolution of semantic conflicts. Naiman and Ouksel have classified semantic conflicts in a similar manner.[6]

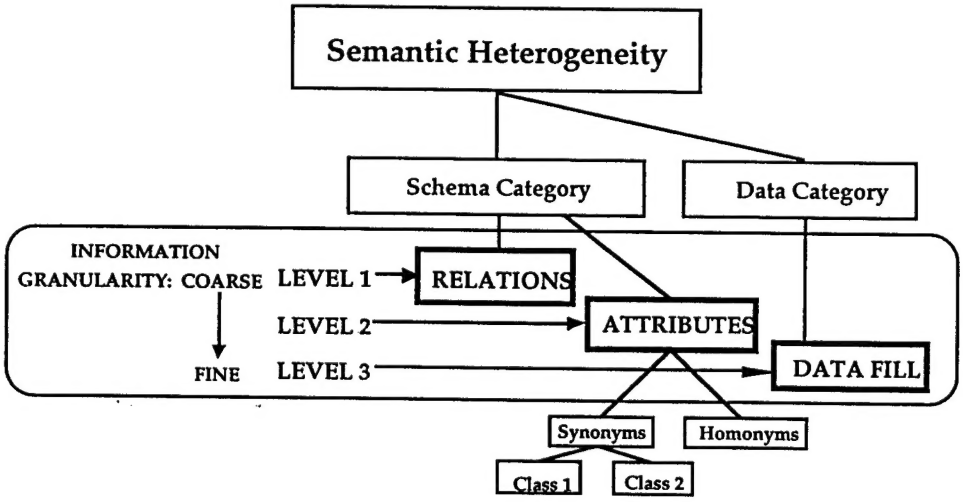

Fig. 1. Categories of semantic heterogeneity and levels of information granularity in database integration.

Levels of abstraction or granularity pertain to objects and also to the information describing them. For example, levels of object abstraction in semantic heterogeneity pertain to physical or notional entities, such as a fleet of ships (coarse), and individual ships (fine). The large, rounded box in Fig. 1 depicts categories of semantic heterogeneity arranged according to information granularity. For example, conflicts in the data category arise from differences in data values returned by similar queries against different databases with attributes representing the same objects. Kamel has presented a detailed classification of semantic conflicts with examples from Naval administrative databases.[7]

2.1. *Level one granularity – relations*

Relations (or "relation variables") [8] are database components at the most coarse-grained level of information. This level is limited to names and definitions of relations, both in comparison to the names and definitions of other relations as well as in comparison to those of attributes. The resolution of semantic inconsistencies at level one does not require access to the data fill. Relation-attribute homonyms occur when a relation and an attribute have the same name. To avoid ambiguity, all relations and attributes should have unique names.

2.2. *Level two granularity – attributes*

The attribute level of granularity includes data-element names, definitions, meanings, data types and lengths. For example, a synonym occurs when the same real-world entity is named differently in different databases. Analysis at level two can resolve semantic conflicts to produce unique attribute names, definitions, data types and lengths for entities in a global, integrated schema. Attributes with different representations in the local schemata that have the same representation in the global schema after analysis at level two are called "equivalent attributes" in this paper.

Data-type conflicts occur when equivalent attributes have different data types (e. g., character vs. numeric), particularly when integrating data managed in different Database Management Systems (DBMSs). For example, one DBMS may use an integer type, whereas another may use a numeric type for the same purpose. Similarly, length conflicts occur when equivalent attributes have different lengths. Type conflicts are quite common when dealing with databases designed for different implementations, whereas length and range conflicts are more likely to occur as a result of semantic choices.[7] The risk of synonyms increases if two users adopt vocabularies at different abstraction levels.[9]

2.2.1. *Synonym classes*

Attribute heterogeneity can be broken down further. For example, synonym abstraction can be divided into two classes. [1, 10 and 11] Class-one synonyms occur when different attribute names represent the same, unique real-world object or concept using the same data type, length, and domain. The only differences between these synonyms are the attribute name and possibly the wording but not the meaning of the attribute definition. In contrast, class-two synonyms occur when different attribute names have equivalent definitions but are expressed with different data types or data-element lengths. Class-two synonyms can share the same domain or can have related domains with a one-to-one mapping between data elements.

Bright, et al.[10] have described strong and weak synonyms, the concept of which is similar to the notion of synonym classes described here. Like class-one synonyms, strong synonyms are semantically equivalent to each other and can be used interchangeably. In

contrast, whereas class-two and weak synonyms are semantically similar and can be substituted for each other in some contexts with minimal meaning changes, they cannot be used entirely interchangeably.[10] Class-two synonyms allow for a one-way as well as a two-way interchange. For example, consider two class-two synonymous attributes with different data-element lengths. The shorter data element can fit into the longer field, but not vice versa. Resolution of semantic inconsistencies in class-two synonyms is much more complicated than that in class-one synonyms.

2.2.2. *Homonyms*

A homonym occurs when different objects or concepts (e.g. entities and attributes) are assigned the same name in different component databases. The risk of homonyms generally is higher when the vocabulary of terms is small, whereas the risk of synonyms is higher when the vocabulary of terms is rich.[9] Homonyms can increase the risk that data integrity will be degraded if the attributes have the same data types and lengths because the error-checking software in the DBMS alone is insufficient to disallow join queries that involve such homonymous attributes, thereby resulting in a meaningless return. In a comprehensive, on-line data dictionary derived from the integration of one or more databases, all instances of semantic heterogeneity at level one and most at level two can be discovered by analyzing the results of appropriate queries on the metadata. Homonym analysis at level two frequently can be performed without consulting the data fill. Detecting synonyms is more difficult because the wording of data definitions can vary while the meanings remain identical.

2.3. *Level three granularity – data fill*

Level three, with the finest granularity, is necessary because many semantic conflicts cannot be resolved at the schema level due to incomplete specification of the metadata. Detection of semantic heterogeneity at level three requires access to the data fill to obtain a better specification of the domains of attributes that appear to be equivalent at level two in order to determine if these attributes represent the same or different objects. Semantic conflicts of different domains at this level arise from different units of measure, different levels of precision, and different ranges of allowable values, etc.

Conflict resolution at level three requires an understanding of domains, which are the sets of all allowed data-element values for attributes. The resolution of semantic inconsistencies at the data-fill level has been hampered by the complexity of domain issues. Frequently, the schema is not sufficiently explicit to exclude values that do not belong in the domains. Resolution of semantic inconsistencies at this level can be very difficult. One reason for this is the flawed manner in which the database industry has implemented the relational model, with no requirement for strong typing.[8] Complete domain specification at the schema level is not supported by commercially available DBMSs and rarely is included in the database system design and implementation. Consequently, the exact domain definitions frequently are ambiguous and are rarely obvious from the schema. How-

ever, precise domain definitions are required for the complete resolution of semantic heterogeneity. Strong typing is one way to address this problem, but this can be time consuming and expensive.

Given this constraint, the most comprehensive solution at the data-fill level that is theoretically possible can be achieved only by considering data updates and data implementation, which are outside the scope of this algorithm. Therefore, we offer a partial solution that depends on an assumption necessitated by the lack of strong typing.

2.3.1. *The domain representation approximation*

Strictly speaking, the multiplicity of domains is not necessarily reflected in the data type and definition at the schema level because strong typing generally was not implemented. The domain cannot always be determined completely using only the information retrieved from a query on the data fill, because the query may return only a subset of the allowed values. That not withstanding, a great deal of domain information can be obtained from the data fill in some cases.

The domain representation approximation allows the data fill present in the database to approximate the domain at level three. It states that the data fill is assumed to represent the domains of the attributes sufficiently to permit correct decisions about whether or not the domain of one attribute is the same as that of another attribute. For example, to apply this approximation to metadata in Table 1, the data fill present for the COAFF attribute is compared the fill for the NATIONALITY attribute to determine if the fills are similar enough to have been derived from the same the domain. This assumption was made because of the necessity to compare the domains of attributes that appear equivalent at the conclusion of the analysis at level two.

During data integration, certain attributes from the different databases emerge into groups of synonyms and homonyms because they have common characteristics, such as the same attribute name, definition, etc. (See section 3.) If the domain subsets obtained from queries on the fill from these "comparable" attributes in databases A and B are very similar, the domain representation assumption implies that these subsets were drawn from a common domain. If the domains appear very different, the equivalence of these attributes is called into question.

The domain representation approximation is most valid when the component databases have the following characteristics:

- Definitions of attributes indicate equivalent or similar data usage.
- Many attributes have user-defined data types.
- A large number of tuples is present in the relations involved in semantic conflicts.
- Attributes have finite domains, such that the number of allowed values in each domain is small compared to the number of tuples (thereby increasing the probability that a select-distinct query will sample the entire domain).

When all of the metadata involved in an integration are rich enough to include a complete specification of the domains of attributes, the analysis at level three will not be required. For example, the Naval Warfare Tactical Database (NWTDB), which supports

military C^4I systems, requires explicit specification of units of measure in the attribute names as part of the schema definition and the data dictionary. (See, for example, references 11 and 12.)

3. Semantic Heterogeneity Case Studies

Data and knowledge engineers are concerned with the semantic implications of the integration of data sources and their corresponding metadata. This section presents some examples from C^4I systems' databases.

The Global Command and Control System - Maritime, (GCCS-M) is the result of a comprehensive, C^4I systems integration effort that supports the U. S. Navy, the Marine Corps and the Coast Guard. A significant contribution to GCCS-M and its predecessors comes from NWTDB, which is the standard, authoritative data source for all Naval tactical warfare systems.[11 and 12] Due to the diverse data sets of NWTDB, the database integration necessary to form NWTDB served as a model for the GCCS-M database integration which includes not only data from NWTDB but other databases required to support a wide variety of maritime C^4I applications with diverse DBMSs. These database-integration efforts provided metadata for case studies in integrating data dictionaries and identifying semantic conflicts.

Table 1 presents sample metadata of some NWTDB components. Because the GCCS-M federated database (FDB) resulted from an integration of several different data sources, the GCCS-M data categories are represented explicitly in the NWTDB and also in Table 1. Component databases designated under "DB" represent NWTDB data sources: "GR" – GCCS-M FDB readiness data from GCCS-M ashore; "GT" – GCCS-M FDB track data from GCCS-M ashore; "M" – Modernized Integrated Database (MIDB) from GCCS-M afloat.

Table 1 shows four examples of Synonym-Homonym Groups (SHGs) which are sets of two or more attributes related by synonymy or homonymy or both.[1 and 11] SHGs also can be called "semantically heterogeneous groups." The concept of the SHG was introduced to focus on the common ground and the diversity among the component databases. The SHGs separated in Table 1 by horizontal lines, include both synonyms and homonyms. A more extensive discussion of SHGs can be found in reference 11. Class-one synonyms also are found in Table 1. For example, synonymous attributes, COAFF and NATIONALITY have the same data type, length and domain.

4. Semantic-Conflict Resolution Algorithm

4.1. *Features of the methodology*

In this section, a methodology is described for identifying and resolving semantic heterogeneity using an algorithm with heuristics. Each phase of this algorithm is based on one of the levels of information granularity shown in Fig. 1 (as opposed to the levels of object granularity discussed in section 2.) The algorithm and its associated heuristics are

Table 1. Examples of synonym-homonym groups derived from C^4I data sets in the Naval Warfare Tactical Database

Attribute name	Relation name	Data Type	Data Length	DB*	Attribute Definition
COAFF	BLUE_FORCE	CHAR	2	GT	Country or international affiliation to which the organization owes allegiance.
CTRY_CODE	AIRFIELDS	CHAR	2	M	Country in which airfield is located.
CTRY_CODE	COUNTRY_CODES	CHAR	2	M	Code assigned to a geographic political area, region or country.
FLAG	SORTSM_ORGLOCN	CHAR	1	GR	Organic resource flag to indicate that reporting unit established subordinate reporting units from its own resources.
FLAG	TRKID	CHAR	2	GT	Code designating country, registry, or political entity to which the platform or unit belongs.
NATIONALITY	UNIT_MASTER_REF.	CHAR	2	GR	Nationality.
FLEET_ID	IDBU	CHAR	1	M	Naval fleet to which a unit is assigned.
FLT	FLEET	CHAR	1	GR	Fleet.
HULL	ESS_MESSAGE_D_E	CHAR	6	GR	Hull number.
HULL	TRKID	CHAR	24	GT	Hull number of ship or submarine, squadron number for fixed-wing aircraft.
HULL_NUMBER	IDBUQL	CHAR	15	M	Hull number of a vessel.
RANK	IDBIND	CHAR	6	M	Rank/grade of an officer within a military service.
RANK	SORTS_UNITCDR	CHAR	4	GR	Rank of commander, the abbreviated rank of the commander, commanding officer, or officer-in-charge.

* DB = Database; GT = GCCS-M Track Database; GR = GCCS-M Readiness Database; M = Modernized Integrated Database

presented in the form of trouble-shooting flow charts, using the hypothetical example of an integration between the local schemata of two component databases, A and B. The objective of the algorithm is to construct a consistent, global, integrated schema for databases A and B that can facilitate data mining and knowledge discovery.

This algorithm can be generalized to apply to the schemata of any number of component databases in a data warehouse and is useful in identifying all of the SHGs present in the aggregate of the component databases. This approach enables data from operational systems to be cleaned periodically and ported into an integrated data warehouse.

The algorithm captured in the flow charts presented as Figs. 2, 3, and 4 was designed to identify and resolve a hierarchy of semantic conflicts, some of which can be resolved by data-dictionary comparison and some of which will require an analysis of the data fill and/or specific domain knowledge at schema-definition time.

In these flow charts, the rectangular boxes represent an action to be performed,. Boxes with bold, rounded corners are used to indicate the starting point at each level. Boxes with bold borders signify a logical transition to the next lower level. The diamonds represent decision points and branches in the procedure. The diamonds with double lines are a reminder that these and other steps need to be performed recursively until all semantic conflicts have been resolved. Plain boxes with rounded corners indicate the end of the procedure or a point at which the analysis should not or cannot continue.

Figs. 2, 3, and 4 describe a systematic procedure designed to ensure that the analyst will not omit inadvertently the comparisons between relations, attributes, and data fill of the component databases. The methodology was designed to resolve semantic inconsistencies at each level before progressing to the next lower level. One proceeds to the next level only when finished at the higher one or when information is needed from a lower level to complete the analysis at the higher one. The flow charts are intended to be applied recursively to the metadata until each instance of semantic heterogeneity is resolved. Thus, the algorithm can be applied to the entire metadata in case all SHGs are not identified, although SHG formation prior to algorithm usage facilitates efficiency of the algorithm by ignoring attributes that do not exhibit semantic inconsistencies. The methodology is designed to eliminate from further consideration metadata irrelevant to semantically related groups, such as SHGs.

4.2. *The hypernym-hyponym group as a mechanism for conflict resolution*

Fig. 3 refers to hypernyms and hyponyms in homonym resolution. The hypernym of a word is defined as a term with a broader, more general meaning, whereas the hyponym of a word expresses the opposite relationship, signifying a more specific meaning.[10] The best way to understand hypernymy and hyponymy is by example. Consider words, A and B. A is a hyponym of B and B is a hypernym of A if any of the following relationships exist between A and B: A "is-a" B; A is "part-of" B; A is a "member-of" B; or A is a "form-of" B.[10 and 13] When this example is applied to attributes in a relation, the domain of A will be a subset of the domain of B in this algorithm. Note that if A is a part of B, the domains of A and B may not be the same or directly related. For example an engine is a part of a ship; however, engines and ships do not share the same or a related domain. The algorithm addresses neither this type of hypernymy nor meronymy ("has a") relationships. (See, for example, reference 13.)

The "many-to-many" relationship between hypernyms and hyponyms implies that a single hyponym could belong to one or more hypernyms. Similarly, a hypernym could have many hyponyms associated with it. Although both verbs and nouns can be hypernyms,[10] the discussion below is limited to noun hypernymy that corresponds to the attributes of a relation. Semantic conflicts involving homonyms and synonyms can arise

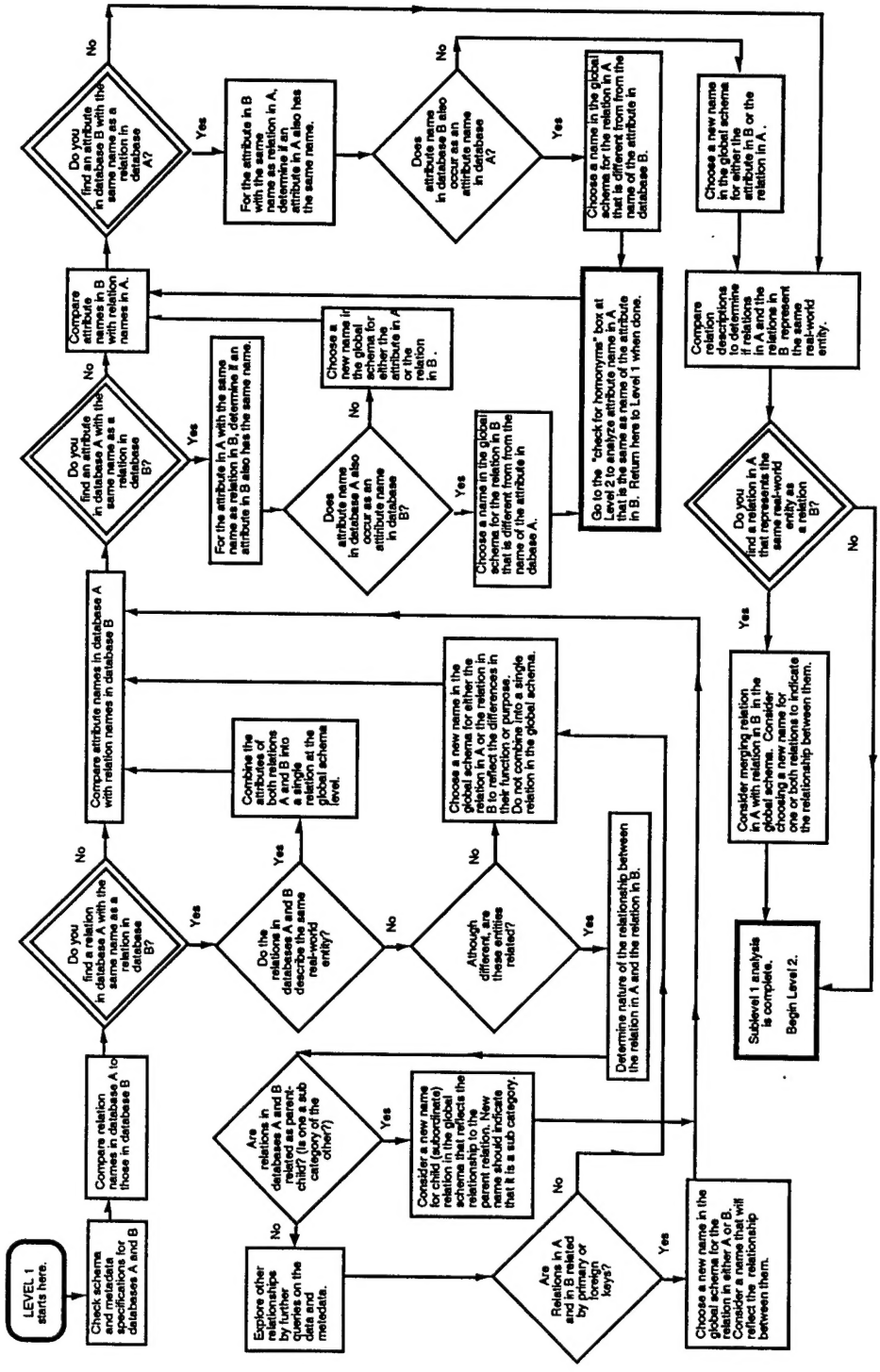

Fig. 2. Trouble-shooting flow chart for level one, relations

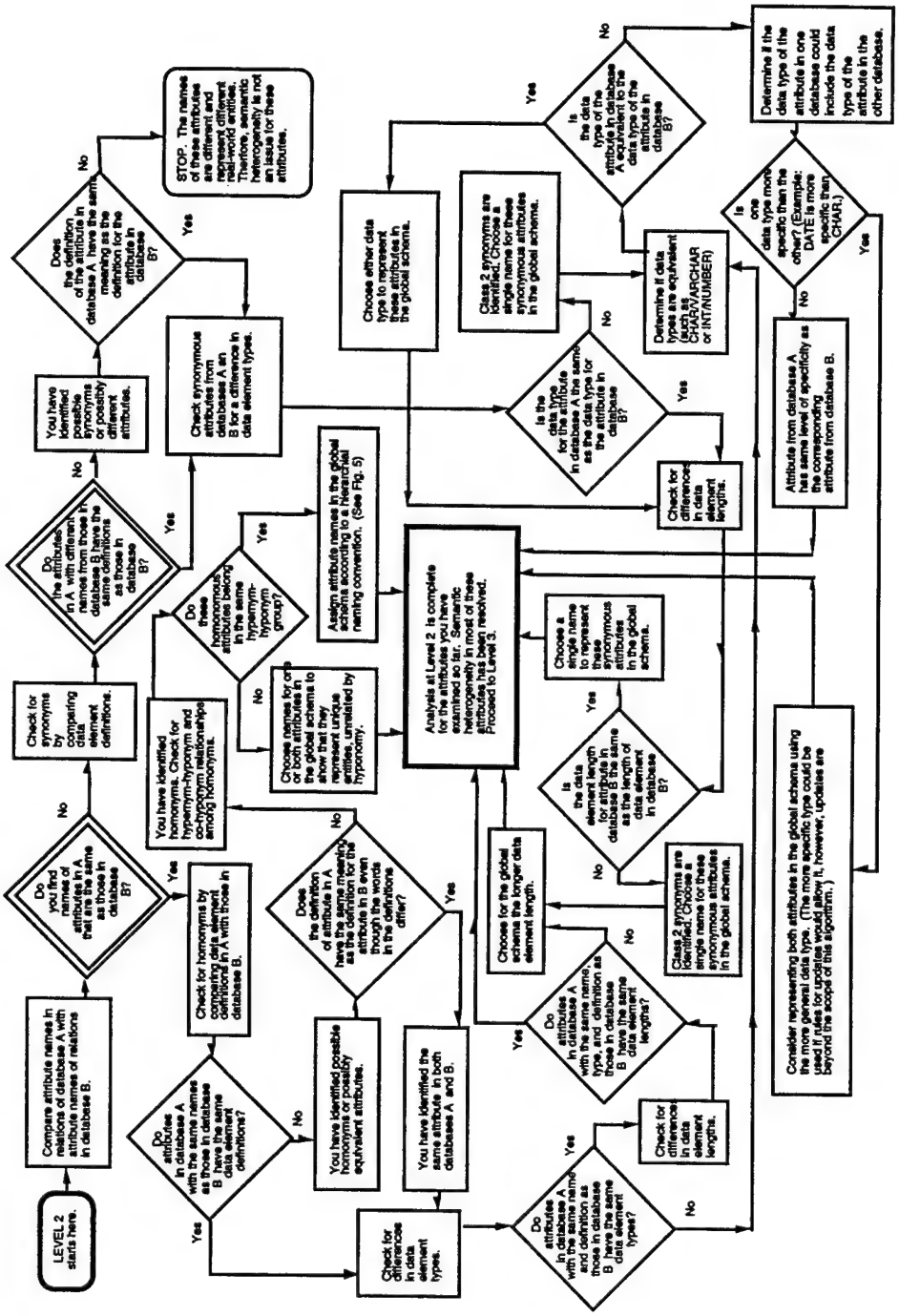

Fig. 3. Trouble-shooting flow chart for level two, attributes

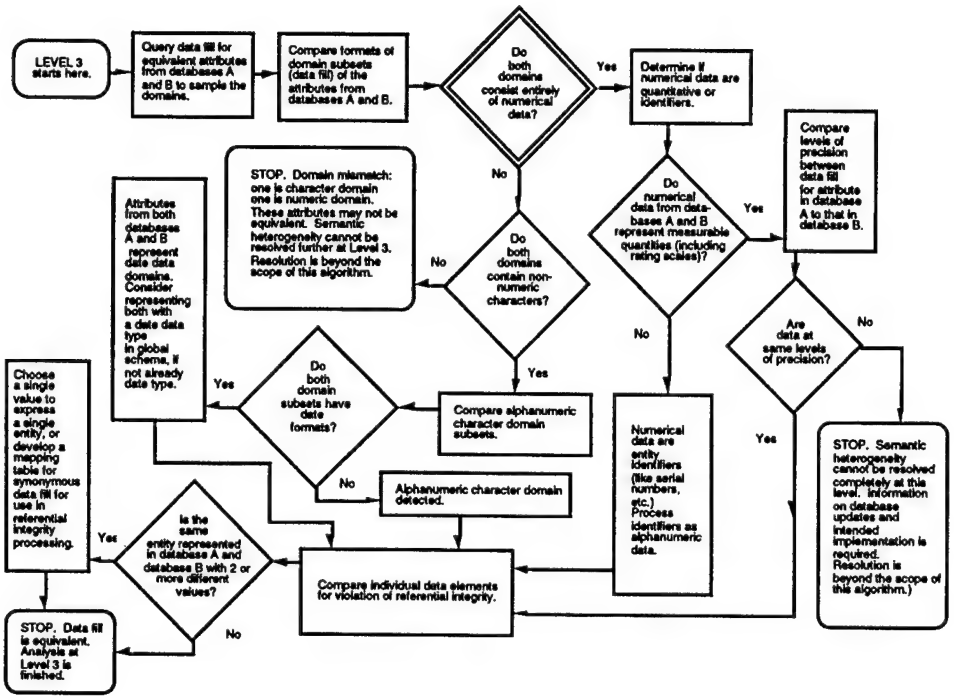

Fig. 4. Trouble-shooting flow chart for level three, data fill

in the development of an integrated data warehouse, even though such issues are not necessarily present in the local schemata. Here, the concept of the Hypernym-Hyponym Group (HHG) is introduced as an aid to resolve some of these conflicts in a logically ordered manner. An HHG is defined as a group of two or more attributes related by hypernymy (and inversely, by hyponymy.) For example, in a manner similar to the formation of SHGs, the metadata in Table 2 are divided into two HHGs consisting of superset-subset collections of attributes. These HHGs also can be combined to form a third HHG, as illustrated in Fig. 5, which is a graphic representation of the metadata in Table 2 that depicts the order among these related attributes.

In general, the formation of well-designed HHGs and the semantically similar hierarchies they represent can offer a more logically organized structure as an aid to the resolution of synonymous, homonymous, and domain-related semantic heterogeneity than a resolution resulting from arbitrarily changing the name or definition of an attribute simply to remove the heterogeneity with no regard to other semantic relationships. The relationships between similar attributes originating from different component databases sometimes are best expressed in terms of HHGs. To specify an HHG completely, the designation of the relative position within the hierarchy of each component must be included. In Table 2 and Fig. 5, the attribute names and definitions clearly provide this information.

Table 2. Hypernym-hyponym groups derived from C^4I data sets in the Naval Warfare Tactical Database

Attribute name	Relation name	Data Type	Data Length	DB*	Attribute Definition
CTRY_CODE	COUNTRY_CODES	CHAR	2	M	A code assigned to a geographic political area, region or country by the Defense Intelligence Agency.
CTRY_CODE_-MFGD	Occurs in 21 tables.	CHAR	2	ND	The country in which the designated system is manufactured.
CTRY_CODE_-MFGD_GUN	GUN_SYSTEM	CHAR	2	ND	The country in which the gun component of the gun system is manufactured.
CTRY_CODE_-MFGD_MOUNT	GUN_SYSTEM	CHAR	2	ND	The country in which the mount component of the gun system is manufactured.
CTRY_CODE_-MFGD_SYS	GUN_SYSTEM	CHAR	2	ND	The country in which the gun system is manufactured.
CTRY_CODE	COUNTRY_CODES	CHAR	2	M	A code assigned to a geographic political area, region or country by the Defense Intelligence Agency.
CTRY_CODE_-USER	Occurs in 15 tablees.	CHAR	2	ND	The country known or estimated to be operating or maintaining a system, weapon, or platform within its inventory.
CTRY_CODE_-USER_ACFT	Occurs in 4 tables.	CHAR	2	ND	The country known or estimated to be operating or maintaining a system, weapon, or platform within its inventory. (Country code of user)
CTRY_CODE_-USER_SUBMERS	Occurs in 4 tables.	CHAR	2	ND	A two-character code assigned to an independent nation-state known or estimated to be operating or maintaining a submersible class from aboard a specific ship, ship class or submarine class. (Country code for submersibles)

* DB = Database; ND = Naval Intelligence Database; M = Modernized Integrated Database

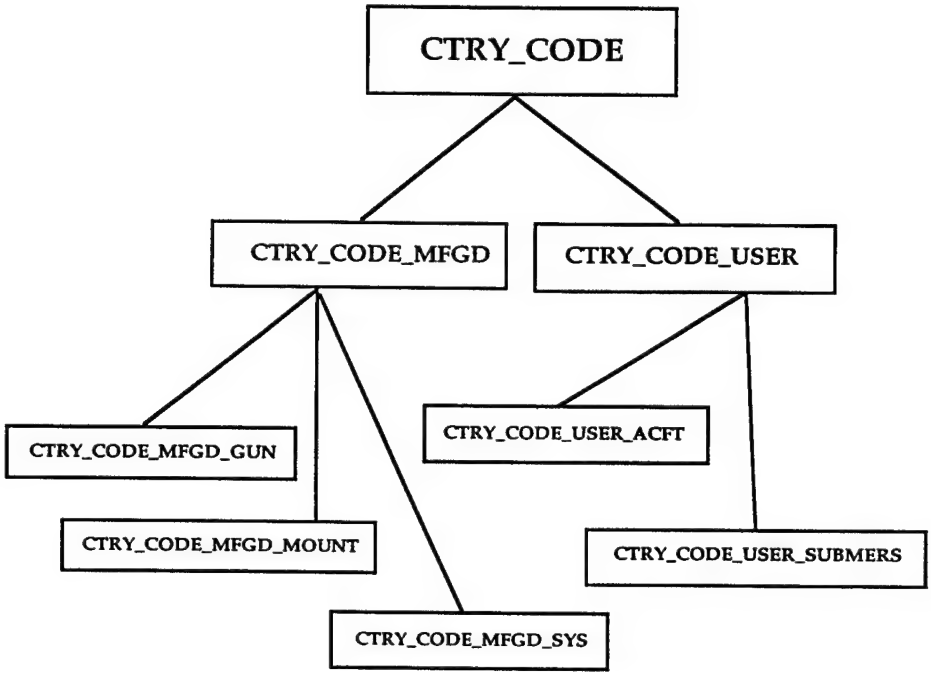

Fig.5. Hypernym-hyponym group structure for the attributes in Table 2.

Table 1 provides an example in which the attribute, RANK, occurs as a homonym from two component databases of the NWTDB. This conflict could be resolved by forming an HHG consisting of one hypernym and one hyponym.

The hypernym, RANK from MIDB, would retain its original name in the global, integrated schema, whereas the hyponym, RANK from the GCCS-M Readiness Database, would be renamed RANK_COM to designate specifically the commander's rank. The attribute definitions would remain the unchanged.

Semantic relationships within HHGs can be characterized further. For example, within HHGs, co-hyponyms are defined as hyponyms having the same, immediate nearest-neighbor parent hypernym. Co-hyponyms have the same level of semantic specificity in the hierarchy. For example, in Fig. 5, CTRY_CODE_MFGD and CTRY_CODE_USER are co-hyponyms because their parent hypernym, CTRY_CODE, is the same for both. Similarly, CTRY_CODE_MFGD_GUN, CTRY_CODE_MFGD_MOUNT, and CTRY_CODE_MFGD_SYS constitute a group of co-hyponyms because they share a common, parent hypernym, namely CTRY_CODE_MFGD. However, although CTRY_CODE_USER_SUBMERS and CTRY_CODE_USER_ACFT are co-hypernyms of each other, neither is considered a co-hypernym of CTRY_CODE_MFGD_GUN because this would violate the requirement for a nearest-neighbor parent hypernym. That not withstanding, CTRY_CODE_MFGD_SYS and CTRY_CODE_USER_ACFT still

have the same level of specificity because they are the same semantic distance from the common, top-level hypernym, CTRY_CODE.

Each hyponym usually will inherit the data type from its parent hypernym. Therefore, all attributes in the same HHG, including all co-hyponyms, frequently will have the same data type. This is the case in the HHG illustrated in Fig. 5, which consists entirely of attribute names and the relationships between them. If a hyponym does not inherit exactly the same data type from its hypernym, the data type of the hyponym will specify a subset of the data type of the hypernym. For instance, a hyponym with data type, "INTEGER" can be related to a hypernym with data type "REAL," since the integers form a subset of the REAL numbers. This also reflects the fact that data types are developed by DBMS vendors to specify certain data domains. Thus, hyponyms can inherit part or all of the domain of the parent hypernym. (The HHGs that the algorithm explicitly addresses consist of only of hyponyms that derive their domains from the top parent hypernym.)

During conflict resolution at level three, domains are compared for homonyms to ascertain if they belong in the same HHG. Because the domain of a hypernym must be a superset of the domains of all its hyponyms, HHGs are formed using concepts very similar to object inheritance[8] in object-oriented design. This relationship between the domains of hypernyms and hyponyms further reinforces the idea that a domain in the relational model is the same as an object class in object-oriented design.[14] For example, all attributes in Table 2 have CHARACTER data types, where the domain of CTRY_CODE_MFGD_SYS is a subset of the domain of CTRY_CODE_MFGD.

In contrast, co-hyponyms will not necessarily share the same domain, because the domains of all hyponyms in the same HHG are required to be subsets only of the domain of the parent hypernym and not of each other. For example, in Fig. 5, CTRY_CODE_MFGD and CTRY_CODE_USER could have the identical domains, domains that intersect each other, or mutually exclusive domains, depending on the relationship between the countries that manufacture equipment and those that use the equipment. The domains of CTRY_CODE_MFGD and CTRY_CODE_USER, however, are required to be subsets of the domain of CTRY_CODE. Finally co-hyponyms must have different definitions; otherwise, they would be synonyms.

4.3. *Implied hypernyms*

Table 3 depicts the concept of an implied hypernym, which is a virtual attribute that constitutes a generalized superset of a group of co-hyponyms. If a hypernym is absent, a group of co-hyponyms can imply that a new hypernym could be created. The relation name is left blank, because these virtual attributes are not actually found in any component database, otherwise they would be actual, as opposed to implied, hypernyms. An example of an implied hypernym (in parentheses), ALT_MAX_FT, together with its co-hyponyms, is shown in Table 3. The name, "ALT_MAX_FT," was chosen for the implied hypernym because it had not been selected previously to name an attribute in NWTDB and because of its descriptive characteristics.

Table 3. Examples of hypernym-hyponym group with implied hypernym (in parentheses) derived from C^4I data sets in the Naval Warfare Tactical Database

Attribute name	Relation name	Data Type	Data Length	DB	Attribute Definition
(ALT_MAX_ FT)		NUM-BER	7		Maximum altitude, in feet, of a weapon-target scenario.
ALT_TGT_FT	AAM_RANGES	NUM-BER	7	ND	Altitude, in feet, of the target
ALT_TGT_MAX_FT	AAM	NUM-BER	7	ND	Maximum altitude, in feet, against which a missile can be expected to be effective for specific range criteria.
ALT_LCH_MAX_FT	AAM	NUM-BER	7	ND	Maximum altitude, in feet, at which a missile can be launched and still function as designed.
ALT_WPN_MAX_FT	GUN_SYSTEM	NUM-BER	7	ND	Maximum altitude, in feet, at which the weapon can engage the target.

Implied hypernyms can be created in HHGs at any level of abstraction in a manner analogous to the formulation of levels in an ontology. The number of implied hypernyms in a single HHG is not restricted.

Implied hypernyms could be identified as such and included in appropriate locations in the schema when the structure of new versions of component databases require the addition of hypernyms as actual attributes before storing the new version in the data warehouse. Implied hypernyms, when included in a global, integrated schema, also can assist database knowledge engineers and users with grasping the relationships between the existing attributes in an integrated data warehouse. With this knowledge, engineers will be in a better position to introduce new attributes or to propose modifications to an existing information structure, while preserving the logical order. The use of implied hypernyms can contribute to better standardization of data-element names.

4.4. *Example of algorithm application*

The following application of some of the heuristics in the algorithm illustrates a relatively simple example of the identification and resolution of semantic heterogeneity in the ship-identifier SHG listed (third from top) in Table 1. The flow charts in Figs. 2, 3, and 4 were designed for a two-component integration; however, they were applied to semantic conflicts in a three-component integration consisting of databases GR, GT and M.

The algorithm can be applied to all metadata at the relations level to generate the SHGs by conducting pairwise comparisons between all relation names and attribute names in the three databases. (Some minor details of the procedure have been omitted for brevity in this example.) Fig. 2 shows all the operations that pertain to relations in this algorithm. It also contains some operations involving attributes as they compare to relations. The following heuristics were extracted from the flow charts in Figs. 2, 3, and 4. Each heuristic is followed by an observation concerning the result of its application.

- Compare the names of the relations in databases GR, GT and M. All relation names are unique.
- Compare names of relations to names of attributes. All three relation names differ from all three attribute names.
- Compare relation descriptions. Whereas examples of relation descriptions are not included in this paper, an analysis of the relation descriptions shows that the relations were designed for unique purposes. Thus application of this heuristic reveals no semantic inconsistencies at the relations level.
- Continue analysis at the attribute level.
- Compare attribute names in database GR to those in databases GT and M, etc. HULL occurs in two of the three databases.
- Compare attribute definitions. HULL has different definitions in databases GR and GT; thus, they are homonyms.
- Compare meanings of attribute definitions. HULL in database GR has a definition equivalent to HULL_NUMBER in database M.
- Compare data-element types and lengths. The application of this heuristic reveals same data type, but different lengths. Thus, HULL in database GR and HULL_NUMBER are class-two synonyms.
- Continue analysis at the data-fill level.

In general, the algorithm seeks to identify and resolve inconsistencies at the higher level before proceeding to the next level, but sometimes information from the data-fill level is required to complete analysis at the attribute level.

- Compare domains of data fill for HULL-related attributes in all three databases. The domains for HULL and HULL_NUMBER are the same in databases GR and M, respectively. This domain is a subset of the domain for HULL in database GT.
- Return to the attribute level.

Now that the semantic conflicts are identified using the information obtained at the data-fill level, the algorithm returns to the attribute level to resolve the inconsistencies.

- Rename HULL in database schema GR and HULL_NUMBER in database schema M. The new attribute name is "HULL_VESSEL."
- Change the attribute definition in database GR to "Hull number of a vessel."
- Increase the length of the "HULL_VESSEL" attribute in database schema GR from 6 to 15 characters.

Semantic heterogeneity is identified and resolved at the attribute level. Table 4 shows the results of the algorithm's application in which the semantic heterogeneity in the ship-identifier SHG from Table 1 has been resolved. Table 4 displays the modified metadata in bold italics. The resolution of semantic conflicts in SHGs can make the order among relations and attributes more apparent. For example, the metadata in Table 4 form an HHG in which HULL is a hypernym and HULL_VESSEL is a hyponym.

Table 4. Processed ship-identifier metadata from Table 1 with semantic conflicts resolved

Attribute name	Relation name	Data Type	Data Length	DB*	Attribute Definition
HULL_VESSEL	ESS_MESSAGE_D_E	CHAR	***15***	GR	***Hull number of a vessel.***
HULL	TRKID	CHAR	24	GT	Hull number of ship or submarine, squadron number for fixed-wing aircraft.
HULL_VESSEL	IDBUQL	CHAR	15	M	Hull number of a vessel.

* DB = Database; GT = GCCS-M Track Database; GR = GCCS-M Readiness Database; M = Modernized Integrated Database. Modified metadata are displayed in ***bold italics.***

4.5. *Limitations of the methodology*

The boundaries of the algorithm are not always distinct because of the complexity and ambiguity in the problem to be solved, particularly at level three, where the heuristics can be less general and less obvious. Some limitations arise from the assumptions and approximations that were made in order to resolve conflicts beyond the schema level. These assumptions will limit the applicability of this method. For example, to implement this methodology, complete, correct, and clearly defined metadata must be available. This is usually, but not always, true of the databases that support major military systems. If no metadata are available, they must be generated through a labor-intensive process that can require much analysis and commitment from the organization sponsoring the work.

The methodology includes decisions about resolving semantic heterogeneity that are somewhat arbitrary because of the arbitrary nature in which many attribute and relation names and definitions are selected in autonomous databases. Moreover, the manner in which attributes and data fill are separated in the autonomous databases also is arbitrary.

Table 5. Examples of the same information with different schemata and fill

System A Relation name: RDYACFT			System B Relation name: MAINTSCHED		
MODEL (N)	AVAILTIME	QTY (Q)	RDYTIME	F15S (Q(N=Y))	F16S (Q(N=Z))
F15 (Y)	0500	22 (W)	0500	22 (W)	—
F16 (Z)	1700	16 (X)	1700	—	16(X)

Attributes and fill are annotated (e.g., N, Q, W, X, Y, Z, (Q(N=Y), etc.) for reference in the text.
Reproduced with permission.[15]

For example, Renner and Scarano showed that two different relations describing the same "real-world" entity and containing the same information will have different schemata and fill plans if the data fill for one relation is part of the schema in the other relation.[15] The problem is illustrated in Table 5, which shows that in system A, the schema and fill plan for a particular relation require metadata item, N, to be stored as an attribute with data fill, Y and Z. Similarly, this relation has another attribute, Q, with data fill, W and X. In contrast, in the schema and fill plan for a different relation describing the same

"real-world" entity and containing the same information in system B specify that the attributes, Q(N=Y) and Q(N=Z), will have fill W and X, respectively, depending on the ready time. Detection and resolution of this type of attribute/data-fill heterogeneity will become increasingly challenging as the numbers of attributes and tuples increase.

The methodology is predicated upon the assumption that an analyst can judge whether data entities are the same or different. Sometimes the context is ambiguous, particularly with class-two synonyms if they cannot be resolved at the data-fill level. Analysis at this level is the most difficult because knowledge of data updates and implementations may be required for the resolution of some data-type heterogeneity. For example, if an attribute requires a numerical data type, a format error could result from an update to the attribute if the allowed data-type requirement has been relaxed to the more general character data type.

HHGs are considered in the flow charts only at the attribute level, for homonym resolution. Actually, HHGs could be formed in the integrated schema from other attributes that are not homonyms if the data-element definitions are related but are not identical. HHGs can be used as a tool to aid an engineer or an analyst in the detection of appropriate cases in which to create implied hypernyms. However, the algorithm is not designed to generate names for implied hypernyms.

This methodology covers several properties of relations, attributes and their data fill. Heterogeneity with respect to nullness; differences in levels of security; data updates; and some kinds of data granularity, except at the relations level, were ignored. Limitations that pertain to the data-fill level are included in Fig. 4. The methodology can report character-numerical domain mismatches, but it cannot resolve them without the input of a data analyst or the use of knowledge-based techniques. Similarly, heterogeneity due to different levels of precision can be discovered but not resolved at the data-fill level.

An implicit assumption during the implementation of this algorithm is that no updates or modifications of any aspect of the component databases will be allowed because these changes could interfere with conflict discovery and resolution.

Whereas this paper is intended to establish a framework for the systematic resolution of semantic inconsistencies, more work is needed in this area, especially to address conflicts arising from data updates and intended use. Because of the variety and complexity of semantic problems, this methodology is appropriate for detecting and resolving some, but not all semantic inconsistencies. For example, although the algorithm can identify class-two synonyms, a better way to resolve them is needed. Finally, the algorithm's performance is expected to degrade in the limit of large data warehouses.

5. Conclusion and Directions for Future Research

A heuristics-based algorithm has been developed for the detection and resolution of semantic conflicts. The approach is explained and illustrated with examples from operational military C^4I-system databases. This method can be applied to cleaning and integrating data sources prior to archiving them in data warehouses to support data mining and knowledge discovery.

In general, the algorithm can be expanded and applied to a wider variety of database-integration situations that cover more cases of semantic heterogeneity in various application domains beyond that of DOD command and control. It can be refined through continued usage and improvements can be made from of experience.

Various DOD agencies increasingly are implementing data warehouses to support management decisions and knowledge discovery regarding business rules and practices. This algorithm can be applied to clean non-tactical as well as tactical data for DOD data warehouses.

The algorithm addresses attribute/relation heterogeneity, but ignores attribute/data-fill heterogeneity, an example of which appears in Table 5. A systematic analysis of the problem of attribute/data-fill heterogeneity also may expose limitations in the degree to which some schemata can be integrated. This is a topic for a separate investigation.

The domain representation approximation and the conditions under which it is a good assumption can be explored quantitatively using techniques based on probability theory. This is related to the uncertainty that Tseng, et al. encountered with queries against heterogeneous databases.[16]

This methodology originally was developed for relational databases; however, it could be modified for an object-oriented data model, where semantic heterogeneity with respect to different object classes and their names could be considered.

The extent to which HHGs, implied hypernyms and co-hyponyms can be used in the construction of integrated schemata and in the integration of data warehouses with knowledge bases is a subject for further research.

Although the algorithm does not involve explicit artificial-intelligence techniques, the heuristics could be captured as axioms in an automated rule-based tool to aid database-integration engineers. However, the resolution of all semantic conflicts cannot be automated completely, especially in the case of legacy data systems for which documentation may be incomplete, incorrect or unavailable. Database-integration tasks require an engineer or analyst to evaluate some data conflicts and formulate solutions based on familiarity with the semantics of the application domain and intended implementation.

Acknowledgments

The authors thank the Space and Naval Warfare Systems Command Information Management Systems Program, the Naval Postgraduate School Institute of Joint Warfare Analysis, and the Defense Advanced Research Projects Agency for their support of this work. This paper is approved for public release with an unlimited distribution.

References

[1] M.G. Ceruti and M.N. Kamel, *Heuristics-based algorithm for identifying and resolving semantic heterogeneity in command and control federated database systems*, Proc. IEEE Knowledge and Data Engineering Exchange Workshop, KDEX'98, Taipei, Taiwan, (Nov. 1998) 17-26.

[2] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth.. *Knowledge discovery and data mining: Towards a unifying framework*, Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, Portland, OR, (Aug. 1996).

[3] D. Hufford, *Data warehouse quality: Part I*, Data Management Review **6**, 1 (1996) 51-53.

[4] M.K. Wysong, *Metadata: Key to successful data warehouse projects*, Proc. 13th Annual DOD Database Colloquium '96, AFCEA, San Diego, CA, (Aug. 1996) 125-132.

[5] A.P. Sheth and J.A. Larson, *Federated database systems for managing distributed, heterogeneous and autonomous databases*, ACM Computing Surveys **22**, 3 (1990) 183-236.

[6] C.E. Naiman and A.M. Ouksel, *A classification of semantic conflicts in heterogeneous database systems*, J. Organizational Computing **5**, 2 (1995) 167-193.

[7] M. N. Kamel, *Identifying and resolving semantic conflicts in distributed heterogeneous databases,* Proc. Tenth Annual DOD Database Colloquium '93, AFCEA, San Diego, CA, (Aug. 1993).

[8] H. Darwen and C.J. Date, *Introducing the third manifesto*, Database Programming and Design **8**, 1 (1995) 25-35.

[9] C. Batini, S. Certi, and S.B. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*, Benjamin/Cummings Publishing Co. (1992).

[10] M.W. Bright, A.R. Hurson and S. Pakzad, *Automated resolution of semantic heterogeneity in multidatabases*, ACM Trans. on Database Systems **19**, 2 (1994) 212-253.

[11] M.G. Ceruti and M.N. Kamel, *Semantic heterogeneity in database and data dictionary integration for command and control systems*, Proc. 11th Annual DOD Database Colloquium '94, San Diego, CA, (Aug. 1994) 65-89.

[12] M.G. Ceruti, S.D. Rotter, K. Timmerman, and J. Ross, *Operations Support System (OSS) integrated database (IDB) design and development: Software reuse lessons learned,* Proc. Ninth Annual AFCEA Database Colloquium '92, (Aug. 1992).

[13] L. Crow and N. R. Shadbolt, *IMPS - Internet Agents for Knowledge Engineering,* Proc. 11th Workshop on Knowledge Acquisition, Modeling and Management , KAW'98, Banff, Canada, http://ksi.cpsc.ucalgary.ca/KAW/KAW98/crow/index.html (Apr. 1998).

[14] P. Drew, R. King, D. McLeod, M. Rusinkiewicz and A. Silberschatz, *Report of the workshop on semantic heterogeneity and interoperation in multidatabase systems*, SIGMOD Record **22**, 3 (1993) 47-56.

[15] S.A. Renner and J.G. Scarano, *Migrating legacy applications to a shared data environment*, Proc. 13th Annual DOD Database Colloquium '96, AFCEA, San Diego, CA, (Aug., 1996) 419-428.

[16] F.S.C. Tseng, A.L.P. Chen, and W.P. Yang, *Answering heterogeneous database queries with degrees of uncertainty,* Distributed and Parallel Databases **1**, 3 (1993) 281-302.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1999	Professional Paper

4. TITLE AND SUBTITLE

Preprocessing and Integration of Data from Multiple Sources for Knowledge Discovery

5. FUNDING NUMBERS

PR: CA60

6. AUTHOR(S)

Dr. M. G. Ceruti(1), and Dr. M. N. Kamel(2)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Center(1)
San Diego, CA 92152–5001

Naval Postgraduate School(2)
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENTR

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The explosive growth in the generation and collection of data has generated an urgent need for a new generation of techniques and tools that can assist in transforming these data intelligently and automatically into useful knowledge. Knowledge discovery is an emerging multidisciplinary field that attempts to fulfill this need. Knowledge discovery is a large process that includes data selection, cleaning, preprocessing, integration, transformation and reduction, data mining, model selection, evaluation and interpretation, and finally consolidation and use of the extracted knowledge. This paper addresses the issues of data cleaning and integration for knowledge discovery by proposing a systematic approach for resolving semantic conflicts that are encountered during the integration of data from multiple sources. Illustrated with examples derived from military databases, the paper presents a heuristics-based algorithm for identifying and resolving semantic conflicts at different levels of information granularity.

Published in International Journal on Artificial Intelligence Tools (IJAIT), 1999.

14. SUBJECT TERMS

Mission Area: Command, Control, and Communications
database integration — knowledge discovery
data mining — semantic heterogeneity
data warehousing

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Same as Report

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b. TELEPHONE *(include Area Code)*	21c. OFFICE SYMBOL
Dr. M. G. Ceruti	(619) 553-4068	Code D4221